

Lehrstuhl für
Rechnerarchitektur & Parallele Systeme
Prof. Dr. Martin Schulz
Dominic Prinz
Jakob Schöffeler

Lehrstuhl für
Design Automation
Prof. Dr.-Ing. Robert Wille
Stefan Engels

Einführung in die Rechnerarchitektur

Wintersemester 2024/2025

Übungsblatt 6: Kombinatorische Schaltungen

25.11.2024 – 29.11.2024

0 Digitaler Schaltungssimulator

In ERA wird zur Erstellung und Simulation von Schaltkreisen **Digital** – ein Projekt von Prof. Dr. Helmut Neemann (Duale Hochschule Baden-Württemberg) – verwendet. Die nötigen Dateien können direkt vom Repository unter <https://github.com/hneemann/Digital> heruntergeladen werden. Es ist keine betriebssystem-abhängige Installation notwendig. Für die Ausführung wird ein Java Runtime Environment (mindestens JRE 8) benötigt. Die JAR-Datei kann direkt gestartet werden, eventuell mit Hilfe des Befehls

```
java -jar Digital.jar
```

Lade **Digital** herunter. Prüfe ob ein JRE korrekt installiert ist und starte **Digital**.

1 Wahrheitstabelle

Überprüfe mittels einer vollständigen Wahrheitstabelle die (Un-)Gleichheit folgender Aussagen:

Hinweis: Beachte, dass wir zu Übungszwecken in den folgenden Teilaufgaben beide Notationen aus der Vorlesung verwenden, d.h., wahlweise $+$, \cdot oder \vee , \wedge .

a) $\overline{x+y} = \overline{x} \cdot \overline{y}$

x	y	$x+y$	$\overline{x+y}$	\overline{x}	\overline{y}	$\overline{x \cdot y}$
0	0	0	1	1	1	1
0	1	1	0	1	0	0
1	0	1	0	0	1	0
1	1	1	0	0	0	0
			lhs	rhs		

b) $x + (y \cdot z) = (x + y) \cdot (x + z)$

x	y	z	y · z	x + (y · z)	x + y	x + z	(x + y)(x + z)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

lhs = rhs

c) $f := \neg((\neg x \wedge y) \vee (\neg y \wedge z)) \wedge (\neg x \wedge z)$ ist eine Kontradiktion

x	y	z	a := $\neg x \wedge y$	b := $\neg y \wedge z$	c := a ∨ b	d := $\neg c$	e := $\neg x \wedge z$	f = d ∧ e
0	0	0	0	0	0	1	0	0
0	0	1	0	1	1	0	1	0
0	1	0	1	0	1	0	0	0
0	1	1	1	0	1	0	1	0
1	0	0	0	0	0	1	0	0
1	0	1	0	1	1	0	0	0
1	1	0	0	0	0	1	0	0
1	1	1	0	0	0	1	0	0

KONTRADIKTION

2 Boolesche Algebra

Prüfe oder widerlege die folgenden Aussagen und verwende dazu die Regeln und Gesetze der Booleschen Algebra. Gib dazu bei jedem Beweisschritt die verwendete Regel bzw. das verwendete Gesetz an.

Hinweis: Beachte, dass wir zu Übungszwecken in den folgenden Teilaufgaben beide Notationen aus der Vorlesung verwenden, d.h., wahlweise $+$, \cdot oder \vee , \wedge .

a) $\overline{a \cdot b} + b = 1$

$$\begin{aligned} \text{lhs} &= \overline{a \cdot b} + b \\ &= \overline{a} + \overline{b} + b \\ &= \overline{a} + 1 \\ &= 1 \end{aligned}$$

1 De Morgan
1 Komplementierung
1 Extremengesetz

lhs = rhs

$$b) \overline{(x+y)} + \overline{(\bar{x}+y)} = \overline{(x+y)} \cdot \overline{(\bar{x}+y)} \quad \text{De Morgan}$$

$$= \overline{(x+y)} \cdot \overline{(\bar{x}+y)} \quad \text{Involution}$$

$$= (x+y) \cdot (\bar{x}+\bar{y}) \quad \text{Kombination}$$

$$= y \quad \text{Kombination}$$

$l_{hs} \neq r_{hs}$

c) $f := \neg((\neg x \wedge y) \vee (\neg y \wedge z)) \wedge (\neg x \wedge z)$ ist eine Kontradiktion.

d) Was ist der duale Ausdruck von $f = \neg((\neg x \wedge y) \vee (\neg y \wedge z)) \wedge (\neg x \wedge z)$. Was können wir über diesen dualen Ausdruck aussagen?

$$f^D = \neg((\neg x \vee y) \wedge (\neg y \vee z)) \vee (\neg x \vee z)$$

$$f = \text{Kontradiktion}$$

$$f^D = 0^D = 1 \quad \text{Tautologie}$$

3 Funktionale Vollständigkeit

Eine Menge $F \subset \mathcal{F}$ an Booleschen Funktionen heißt **funktional vollständig** wenn jede Boolesche Funktion sich als Komposition von Funktionen $f \in F$ sowie den Variablen x_1, \dots, x_n schreiben

lässt. Man beachte, dass auch die Konstanten 0 und 1 als Funktionen in \mathcal{F} aufgefasst werden, nämlich diejenigen, die alle Variablenbelegungen konstant auf 0 oder 1 abbilden. Man kann zwar in der Praxis davon ausgehen, dass man beim Schaltungsentwurf immer Zugriff auf eine konstante 0- und 1-Leitung hat, in der Theorie der funktionalen Vollständigkeit bleibt dies jedoch unberücksichtigt. In der Vorlesung wurde bereits erwähnt, dass die Menge $\{\wedge, \vee, \neg, 0, 1\}$ funktional vollständig ist.

Zeige oder widerlege die funktionale Vollständigkeit der folgenden Mengen:

Hinweis: Du weißt bereits aus der Zentralübung, dass auch $\{\wedge, \neg\}$ funktional vollständig ist.

- $\{NOR\}$
- $\{\oplus, 0, 1\}$
- $\{\neg, \leftrightarrow\}$
- $\{ITE, 0, 1\}$, wobei $ITE: \{0, 1\}^3 \rightarrow \{0, 1\}$ mit $ITE(x, y, z) := (x \wedge y) \vee (\neg x \wedge z)^1$.

Bonusaufgabe:

Eine Boolesche Funktion f ist *wahrheitserhaltend* wenn der Wert der Funktion 1 ist wenn alle Variablen der Funktion auf 1 gesetzt werden, also $f(1, 1, \dots, 1) = 1$. Zeige, dass die Menge aller wahrheitserhaltenden Funktionen *nicht* funktional vollständig ist.

4 Arithmetische Schaltung

Gegeben sind zwei 2-bit unsigned Integer $a = a_1a_0$ und $b = b_1b_0$. Es soll ein Schaltkreis entworfen werden der den Absolutwert der Differenz $y = y_1y_0 = |a - b|$ berechnet.

- Entwerfe den Schaltkreis unter Verwendung von Halb- und Volladdierern.
- Simuliere den Schaltkreis in Digital.
- Betrachte wie y_0 von den Eingängen abhängt. Fällt dir eine einfache Schaltung ein die y_0 direkt berechnet?

5 Hausaufgabe: Kombinatorische Schaltung: 16-Segment Display

Bearbeitung und Abgabe der Hausaufgabe 6 auf <https://artemis.in.tum.de/courses/401> bis **Sonntag, den 01.12.2024, 23:59 Uhr**.

In dieser Aufgabe soll, basierend auf einer textuellen Beschreibung, eine Schaltung entwickelt werden, die ein 16-Segment-Display ansteuert. Ziel ist es, mithilfe von Logikgattern eine Schaltung zu entwerfen, die das Display steuern kann.

¹ITE wird auch *If-Then-Else-Operator* genannt, da $ITE(1, y, z) = y$ und $ITE(0, y, z) = z$.

3)

a) $\{ \text{NOR} \}$

$$\neg: \text{NOR}(a, a) = \neg(a \vee a) = \neg a$$

$$\wedge: \text{NOR}(\text{NOR}(a, b), \text{NOR}(a, b)) = \neg(\neg(a \vee b) \vee \neg(a \vee b)) = \neg(\neg(a \vee b)) \wedge \neg(\neg(a \vee b)) \\ = (a \vee b) \wedge (a \vee b) = a \vee b$$

b) $\text{ITE}(x, 1, 0)$

$\{ \text{ITE} \}$

```

if x == 1
    return 0
else
    return 1

```

$$\neg: \text{ITE}(x, 0, 1) = (x \wedge 0) \vee (\neg x \wedge 1)$$

$$= 0 \vee \neg x$$

$$= \neg x$$

$$\wedge: \text{ITE}(x, y, 0) = (x \wedge y) \vee (\neg x \wedge 0) \\ = x \wedge y$$

4)

b war größer als a , falls
 $c_1 = 0$

a_1	a_0	b_1	b_0	y_1	y_0	s_0	c_0	s_1	c_1
0	0	0	0	0	0	0	1	0	1
0	0	0	1	0	1	1	0	1	0
0	0	1	0	1	0	0	1	1	0
0	0	1	1	1	1	1	0	0	0
0	1	0	0	0	1	1	1	0	1
0	1	0	1	0	0	0	1	0	1
0	1	1	0	0	1	1	1	1	0
0	1	1	1	1	0	0	1	1	0
1	0	0	0	1	0	0	1	1	1
1	0	0	1	0	1	1	0	0	1
1	0	1	0	0	0	0	1	0	1
1	0	1	1	0	1	1	0	1	0
1	1	0	0	1	1	1	1	1	1
1	1	0	1	1	0	0	1	1	1
1	1	1	0	0	1	1	1	0	1
1	1	1	1	0	0	0	1	0	1

SA invention

SA invention

SA invention

SA invention

Das Display verfügt über einen 16-bit Segmenteingang. Jedes Bit (von 0 bis 15) steuert ein Segment. Wenn ein Bit auf „1“ gesetzt ist, leuchtet das entsprechende Segment, wie in der Abbildung dargestellt.

a) Bestimme, welche Segmente des Displays aktiviert werden müssen, um die Buchstaben E, R und A korrekt darzustellen.

- Hinweis: Es sind insgesamt 6 Gruppen. Eine dieser Gruppe umfasst die Segmente, die bei allen drei Buchstaben aus sind.

- © 2024 Lehrstuhl für Rechnerarchitektur & Parallele Systeme, alle Rechte vorbehalten

A	B	out
0	0	Aus
0	1	E
1	0	R
1	1	A

d) Realisiere die Schaltung in Digital und verwende dabei ausschließlich NAND-Gatter². Die Eingänge der NAND-Gatter dürfen nicht(!) invertiert werden.

- Eingang: Die Schaltung soll einen 2-bit Eingang besitzen, der zwischen den vier Zuständen umschaltet.
- Ausgang: Ein 16-bit Ausgang, der als Eingang für ein 16-Segment Display verwendet werden kann.

Hinweis: Es empfiehlt sich, zunächst eine Schaltung „nur“ für die 6 Segmentgruppen zu entwickeln, da diese sonst schnell unübersichtlich und redundant wird. Mit Hilfe eines Mergers können diese 6 bit dann einfach auf die 16 Ausgangs-bit entsprechend der Überlegungen in Teilaufgabe b verdrahtet werden.

© 2024 Lehrstuhl für
Rechnerarchitektur &
Parallele Systeme
alle Rechte vorbehalten

Beachte für die Abgabe auf Artemis:

- Die bestehenden Elemente dürfen nicht verändert werden, d.h., insbesondere dürfen die Namen und Bit-Breiten nicht angepasst werden. Die Elemente dürfen jedoch in ihrer Position verschoben werden.
- Das Template für Teilaufgabe d beinhaltet auch eine 16-Segment-Anzeige, die zum visuellen Debugging verwendet werden kann. Selbstverständlich wird daher auch die 16-Segment-Anzeige vom Tester als Komponente bei der Abgabe erlaubt.

²Konstanten, Tunnel und Splitter/Merger aus der Untergruppe „Leitungen“ dürfen in Digital immer verwendet werden.